14136

Postel

Network Working Group
Request for Comments # 452
NIC# 14136
Categories: TELNET

Joel Winett, LL (JMW)
February 8, 1973

Rcvd 22FEB73

TELNET Command at Host LL

The attached writeup documents the use of the TELNET command at Host LL for uses under the CP/CMS time-sharing system. This command provides for use in a HALF DUPLEX or FULL DUPLEX mode and can be used with ASCII codes or with EBCDIC codes (i.e., transparent mode). The keyboard conventions allow for entering all ASCII codes using a CONTROL character followed by another character for the codes not present on an IBM 2741 terminal. In addition a number of local TELNET control commands have been defined which allows the mode of operation to be changed, the redefinition of special characters, the sending of TELNET control codes, as well as for obtaining input from a file instead of from the terminal and for sending output to a file instead of to the terminal. These later features provide for a rudamentary file transfer facility.

This material has not been reviewed for public release and is intended only for use with the ARPA network. It should not be quoted or cited in any publication not related to the ARPA network.

TELNET

Purpose:

To access another terminal oriented system on the ARPA network.

Format:

TELNET host <tag> RESUME EBCDIC HALFDUP
1 OPEN ASCII FULLDUP

host - either the hexadecimal code for a foreign network service site or a standard mnemonic for a foreign site. See Figure 1.

tag - the identifier for the local connections to the network. The tag is used together with the address of the virtual machine descriptor table (UTABLE) to form local socket numbers which are used in the network protocol.

RESUME - used to reactivate communications with a foreign site after having previously left the TELNET command leaving the connections open.

EBCDIC - to communicate with EBCDIC codes. The default is network ASCII.

HALFDUP - to operate under a half duplex protocol, i.e. with a locked keyboard.

The EBCDIC HALFDUP the protocol assumes that the TELNET break code (circle C) will be received to indicate when the keyboard should be locked for input.

In ASCII HALFDUP the keyboard will lock after a line of input and will unlock after one or more lines have been received for output. An external interrupt will also unlock a locked keyboard.

The default is full duplex where the keyboard is alwasy unlocked for input. A null line is required to temporarily lock the keyboard in order to receive output

Usage:

A number of hosts on the ARPA network provide TELNET service. A Network Virtual Terminal (NVT) has been specified so that using sites can write one TELNET program which maps a local terminal into the NVT to access any serving site on the network. Once ccmmunication has been established between a using site and a serving site, keyed input is sent to the serving system and output from the serving site, when received, is typed on the local terminal.

The NVT protoccl requires that the keyboard be capable of entering all of the 128 ASCII codes together with a number of other TELNET control codes. To support an NVT with an IBM 2741 terminal, it is necessary to adapt a control convention for entering codes which are not associated with single keys on the keyboard. In addition, since CP/CMS processes input from a 2741 on a line at a time terminated with a newline, a means must be established for entering a sequence of characters for transmission which is not terminated with a newline code.

When TELNET is initiated the message

ENTER CONTROL CHARACTER

is typed. A non-blank character should then be entered which defines the character which, in combination with another character, will be used to enter codes not associated with single keys. The control character is also used for other special control functions as described below.

Codes:

The NVT usually requires that characters be transmitted in an eight bit ASCII code. Since the TELNET command is written to process EBCDIC codes ASCII codes received are translated into EBCDIC and characters to be transmitted are translated into ASCII before being sent tc a serving site. Figure 2 gives the ccmplete definition of EBCDIC indicating the EBCDIC controls and EBCDIC graphics. Figure 3 gives the codes for the ASCII controls and graphics. The complete mapping between 8-bit EBCDIC codes and 8-bit network ASCII codes is shown in Figure 4. The EBCDIC newline code (NL) is mapped into the ASCII codes for the pair of characters CR-IF.

The follwoing ASCII/EBCDIC mapping is used for the non-EBCDIC graphics:

	ASCII		EBCDIC	
TILDE	(7E)	=	(A1)	NOT
BAR	(7C)	=	(6A)	OR
BACK SIASH	(5C)	=	(E0)	
CARAT	(5E)	=	(71)	
GRAVE	(60)	=	(79)	
LEFT BRACE	(7B)	=	(8B)	
RIGHT BRACE	(7D)	=	(9B)	
LEFT BRACKET	(5B)	=	(AD)	
RIGHT BRACKET	(5D)	=	(BD)	

The ASCII control DC3 (X'13') maps to the EBCDIC ccntrol TM (X'13'). The ASCII control NUL (X'00') is sent to the terminal as the EBCDIC code for NULL (X'00') and is not mapped into an IDLE (X'17').

The TELNET control hide-your-input is mapped into the EBCDIC code for bypass (print supress) and the TELNET control noecho is mapped into the EBCDIC code for restore (print restore). If the TELNET control for echo is received, a message is printed and it is mapped into an IDLE. Similiarly, if the TELNET control for break is received, a message is printed and it is mapped into an IDLE unless operation is in EBCDIC HALFDUP mode in which case the break is used to indicate that any received characters should be printed and the keyboard unlocked for input. If a data mark or an interrupt is received, no action is taken except to print a message to notify the user of this occurance.

Input:

When the control character is entered, the following character is mapped into a different code than that which it is normally mapped into, except when the following character is a space or a character not defined to have a meaning when preceded by the control character. Figure 5 gives the mapping of the characters on a 2741 keyboard when preceded by a control character. The following 2741 keyboard characters do not have a different meaning when preceded by the control character.

$ # * % &
+ - = _
. , : ;
! | ? ¢
SPACE
BACKSPACE
TAB

When a character is mapped into its control code, the control character is mapped into the code for IDLE. If the control character is entered as the last character before the newline key is entered, the sequence of characters entered is transmitted without the newline code. That is, the newline code is not transmitted when it is preceded by the control character.

When the 2741 keyboard is unlocked for input, characters received cannot be typed until the keyboard is locked again. After a line is entered, received characters can then be typed. When operating in full duplex or ASCII half duplex, a null line entered will allow received characters to be typed but will not cause the new line code to be transmitted. To cause a null line, i.e., just the new line code to be transmitted, the control character should be entered as the only character in the input line. In EBCDIC HALFDUP a null line entered will cause a null line to be transmitted.

Output:

ASCII output received from the NVT is converted into EBCDIC with the sequences CR-LF converted into IDLE-NL. The EBCDIC characters are then sent to the terminal. Note that not all 128 ASCII codes when converted to EBCDIC will print on a 2741. Of the 95 ASCII graphics and the 8 ASCII controls which are defined for the NVT printer, the following are not visible or audable:

CARAT
GRAVE
BACK SLASH
LEFT BRACE
RIGHT BRACE
LEFT BRACKET
RIGHT BRACKET
ASCII CONTROLL BELL (BEL)
ASCII CONTROL VERTICAL TAB (HT)
ASCII CONTROL FORM FEED (FF)
ASCII CONTROL CARRIAGE RETURN (CR)

Figure 6 shows how the EBCDIC codes from X'40' through X'FF' will appear on a 2741 terminal. Figure 7 shows how the EBCDIC codes will appear when printed with a PN train on the offline printer and Figure 8 shows how these codes appear when printed with a TN train.

Controls:

If the first character in an input line is the control character and the next character is a space, the rest of the line is interpreted as a TELNET control command. A control command consists of a control word and parameters separated by spaces. Controls are defined which permit TELNET controls to be transmitted to the serving site, allow input to come from a file or output to go to a file, allow CMS functions or transient commands to be issued, redefine the control character or TELNET mode, close connections or leave the TELNET command with the connections still open, as well as controls to support a reader, punch, and printer with RJS operation. The controls are described below.

CONTROL x

Where x is the new control character

CLOSE

Tc close all connections and quit

QUIT

To leave TELNET

EBCDIC

To go into transparent mode, i.e., no translation

ASCII

To translate input and output to network ASCII

BREAK

To send the TEINET break code

SYNC

To send the TEINET data mark code and an interrupt

ATTN

To send a TELNET break and a SYNC

HIDE-YOUR-INPUT

To send the TEINET hide you input code.

NOECHO

To send the TEINET noecho code

ECHO

To send the TEINET echo code.

CMS command arg1 . . . argN

To issue CMS core resident function or transient ccmmand.

INPUT fn ft
* TERMIN
* *

To get input from a file If fn is defaulted, input is reset to come from the terminal. If fn is * file input resumes after the last line read. After an EOF, the next line read will be the first line of the file.

An external interrupt while input is coming from a file will cause the line number of the next line to be read from the file to be typed and input to be reset to come from the terminal.

```
OUTPUT fn OFF  TERM   INPUT   INOUT
       *  ON   NOTERM NOINPUT OUTPUT
```

To write output to the file 'fn TERMOUT'. If fn is defaulted, output is reset to go to the terminal. If fn is *, file OUTPUT is resumed with the same options as were last used.

For Output to the Terminal:

If the last character is a CR, a line with just the control character is typed on the next line (with a NL)

If the last character is not a NL or a CR, the line is typed without a NL (i.e., with TYPE).

For Output to a File:

If just a NL is in the line, just the control character is sent to the file.

If the last CHAR is not NL or CR, the control character is added after the last character, except if 130 characters must be sent to the file.

If the last CHAR is a CR, it is included in the file.

OFF causes all output to be discarded.

ON is the default, and causes output to go to the terminal.

TERM causes output to also go to the terminal.

NOTERM is the default, and causes output to go the the file but not to the terminal.

OUTPUT is the default and causes just terminal output to be put to the file 'FN termout'.

INOUT causes both terminal input and terminal output to be put to the cutput file.

INPUT causes terminal input but not output to be put to the output file.

NOINPUT is defaulted and causes input to not go to the file.

PURGE

To purge all output currently received by the NCP.
***** NOT YES IMPLEMENTED *****

READER fn ft
* READER

To send a job to the RJS system at UCLA's CCN.

If fn and ft are defaulted, input will come from the card reader.

PRINTER fn ft
* PRINTER

To receive printer output from the RJS system at UCLA's CCN.

To receive punch output from the RJS system at UCLA's CCN.

If fn and ft are defaulted, output goes to the printer.

PUNCH fn ft
* PUNCH

If fn and ft are defaulted, output goes to the punch.

HOST	SITE	MACHINE	SYSTEM	HOST NUMBER DEC	OCT	HEX
NMC	UCLA	SIGNA-7	SEX	1	1	01
ARC	SRI	PDP-10	NIC	2	2	02
UCSB	UCSB	360/75	OS/MVT	3	3	03
UTAH	UTAH	PDP-10	TENEX	4	4	04
MULTICS	MIT	H-645	MULTICS	6	6	06
SDC	SDC	370/155	ADEPT	8	10	08
HARV	HARVARD	PDP-10	4S72	9	11	09
LL	LL	360/67	CP/CMS	10	12	0A
CASE	CASE	PDP-10	10/50	13	15	0D
CMU	CMU	PDP-10	TOPS-10	14	16	0E
ILLIAC	AMES	B-6500	?	15	17	0F
AMES	AMES	360/67	TSS/360	16	18	10
CCN	UCLA	360/91	OS/MVT	65	101	41
SRI	SRI-AI	PDP-10	TENEX	66	102	42
BBNA	BBN	PDP-10	TENEX	69	105	45
DMCG	MIT	PDP-10	ITS	70	106	46
RAND	RAND-FCC	PDP-10	TENEX	71	107	47
TX2	LL	TX-2	APEX	74	112	4A
BBNB	BBN	PDP-10	TENEX	133	205	85
MITAI	MIT	PDP-10	ITS	134	206	86

Serving Hosts on the ARPA Network
FIGURE 1

	0 0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	1 0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	2 0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	3 0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
4567																
0000	NUL	DLE	DS		SP	&	-				-	0				0
0001	SOH	DC1	SOS				/		a	j	°	¹	A	J		1
0010	STX	DC2	FS	SYN					b	k	s	²	B	K	S	2
0011	ETX	TM							c	l	t	³	C	L	T	3
0100	PF	RES	BYP	PN					d	m	u	⁴	D	M	U	4
0101	HT	NL	IF	RS					e	n	v	⁵	E	N	V	5
0110	IC	ES	ETB	UC					f	o	w	⁶	F	O	W	6
0111	DEL	IL	ESC	EOT					g	p	x	⁷	G	P	X	7
1000	GE	CAN							h	q	y	⁸	H	Q	Y	8
1001	RLF	EM							i	r	z	⁹	I	R	Z	9
1010	SMM	CC	SM		¢	!		:								
1011	VT	CU1	CU2	CU3	.	$	,	#	{	}	˪	˩				
1100	FF	IFS		DC4	<	*	%	@	≤	▫	⌐	¬				
1101	CR	IGS	ENQ	NAK	(	)	_	'	‘	’	[	]				
1110	SO	IRS	ACK		+	;	>	=	⁺	±	≥	≠				
1111	SI	IUS	ELL	SUB			¬	?	"	+	▪	•	–			

	0	1	2	3	4	5	6	7

Code Structure

Extended Binary-Coded Decimal Interchange Code (EBCDIC)
FIGURE 2

8 7 6 5 4 3 2 1	8 0 0 0 0 0 0 0	7 0 0 0 1 1 1 1	6 0 1 1 0 0 1 1	5 0 1 0 1 0 1 0 1
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Code Structure

USA Standard Code for Information Interchange (USASCII)
FIGURE 3

ASCII DEC	ASCII OCT	ASCII HEX	SYMBOLS	EBCDIC HEX	EBCDIC DEC
0	0	(00)	NUL	(00)	00
1	1	(01)	SOH	(01)	01
2	2	(02)	STX	(02)	02
3	3	(03)	ETX	(03)	03
4	4	(04)	EOT	(37)	55
5	5	(05)	ENQ	(2D)	45
6	6	(06)	ACK	(2E)	46
7	7	(07)	BEL	(2F)	47
8	10	(08)	BS	(16)	22
9	11	(09)	HT	(05)	05
10	12	(0A)	LF	(25)	37
11	13	(0B)	VT	(0B)	11
12	14	(0C)	FF	(0C)	12
13	15	(0D)	CR	(0D)	13
14	16	(0E)	SO	(0E)	14
15	17	(0F)	SI	(0F)	15
16	20	(10)	DLE	(10)	16
17	21	(11)	DC1	(11)	17
18	22	(12)	DC2	(12)	18
19	23	(13)	DC3	(13)	19
20	24	(14)	DC4	(3C)	60
21	25	(15)	NAK	(3D)	61
22	26	(16)	SYN	(32)	50
23	27	(17)	ETB	(26)	38
24	30	(18)	CAN	(18)	24
25	31	(19)	EM	(19)	25
26	32	(1A)	SUB	(3F)	63
27	33	(1B)	CTL	(27)	39
28	34	(1C)	FS	(1C)	28
29	35	(1D)	GS	(1D)	29
30	36	(1E)	RS	(1E)	30
31	37	(1F)	US	(1F)	31

ASCII/EBCDIC Code Mappings
FIGURE 4

ASCII DEC	ASCII OCT	ASCII HEX	SYMBOLS	EBCDIC HEX	EBCDIC DEC	
64	100	(40)	@	(7C)	124	
65	101	(41)	A	(C1)	193	
66	102	(42)	B	(C2)	194	
67	103	(43)	C	(C3)	195	
68	104	(44)	D	(C4)	196	
69	105	(45)	E	(C5)	197	
70	106	(46)	F	(C6)	198	
71	107	(47)	G	(C7)	199	
72	110	(48)	H	(C8)	200	
73	111	(49)	I	(C9)	201	
74	112	(4A)	J	(D1)	209	
75	113	(4B)	K	(D2)	210	
76	114	(4C)	L	(D3)	211	
77	115	(4D)	M	(D4)	212	
78	116	(4E)	N	(D5)	213	
79	117	(4F)	O	(D6)	214	
80	120	(50)	P	(D7)	215	
81	121	(51)	Q	(D8)	216	
82	122	(52)	R	(D9)	217	
83	123	(53)	S	(E2)	226	
84	124	(54)	T	(E3)	227	
85	125	(55)	U	(E4)	228	
86	126	(56)	V	(E5)	229	
87	127	(57)	W	(E6)	230	
88	130	(58)	8	(E7)	231	
89	131	(59)	Y	(E8)	232	
90	132	(5A)	Z	(E9)	233	
91	133	(5B)	[	(AD)	173	
92	134	(5C)	¢	(4A)	74	(BACK-SLASH)
93	135	(5D)	]	(BD)	189	
94	136	(5E)		(71)	113	(CARAT)
95	137	(5F)	_	(6D)	109	

ASCII/EBCDIC Code Mappings
FIGURE 4 (CONTINUED)

ASCII DEC	ASCII OCT	ASCII HEX	SYMBOLS	EBCDIC HEX	EBCDIC DEC	
96	140	(60)		(79)	121	(GRAVE)
97	141	(61)	a	(81)	129	
98	142	(62)	b	(82)	130	
99	143	(63)	c	(83)	131	
100	144	(64)	d	(84)	132	
101	145	(65)	e	(85)	133	
102	146	(66)	f	(86)	134	
103	147	(67)	g	(87)	135	
104	150	(68)	h	(88)	136	
105	151	(69)	i	(89)	137	
106	152	(6A)	j	(91)	145	
107	153	(6B)	k	(92)	146	
108	154	(6C)	l	(93)	147	
109	155	(6D)	m	(94)	148	
110	156	(6E)	n	(95)	149	
111	157	(6F)	o	(96)	150	
112	160	(70)	p	(97)	151	
113	161	(71)	q	(98)	152	
114	162	(72)	r	(99)	153	
115	163	(73)	s	(A2)	162	
116	164	(74)	t	(A3)	163	
117	165	(75)	u	(A4)	164	
118	166	(76)	v	(A5)	165	
119	167	(77)	w	(A6)	166	
120	170	(78)	x	(A7)	167	
121	171	(79)	y	(A8)	168	
122	172	(7A)	z	(A9)	169	
123	173	(7B)	{	(8B)	139	
124	174	(7C)	|	(4F)	79	(BAR/OR)
125	175	(7D)	}	(9B)	155	
126	176	(7E)	¬	(5F)	95	(TILDE/NOT)
127	177	(7F)	DEL	(07)	7	

ASCII DEC	ASCII OCT	ASCII HEX	TELNET CONTROLS	EBCDIC HEX	EBCDIC DEC	
128	100	(80)	DATA-MARK	(80)	128	
129	101	(81)	BREAK	(38)	56	
130	102	(82)	NOP	(17)	23	IDLE
131	103	(83)	NOECHO	(14)	20	RESTORE
132	104	(84)	ECHO	(23)	35	
133	105	(85)	HIDE-YOUR-INPUT	(24)	36	BYPASS

ASCII/EBCDIC Code Mappings
FIGURE 4 (CONTINUED)

	EBCDIC		EBCDIC	ASCII
CENT	(4A)	= ESC	(27)	(1B)
CTL <	(4C)	= LEFT BRACKET	(AD)	(5B)
CTL >	(6E)	= RIGHT BRACKET	(BD)	(5D)
CTL (	(4D)	= LEFT BRACE	(8B)	(7B)
CTL)	(5D)	= RIGHT BRACE	(9B)	(7D)
CTL /	(61)	= BACK SLASH	(4A)	(5C)
CTL "	(7F)	= CARAT	(71)	(5E)
CTL '	(7D)	= GRAVE	(79)	(60)
CTL 6	(F6)	= FS	(1C)	(1C)
CTL 7	(F7)	= GS	(1D)	(1D)
CTL 8	(F8)	= RS	(1E)	(1E)
CTL 9	(F9)	= US	(1F)	(1F)
CTL _	(6D)	= US	(1F)	(1F)
CTL ¬	(5F)	= DEL	(07)	(7F)
CTL @	(7C)	= NUL	(00)	(00)
CTL A	(C1)	= SOH	(01)	(01)
CTL B	(C2)	= STX	(02)	(02)
CTL C	(C3)	= ETX	(03)	(03)
CTL D	(C4)	= EOT	(37)	(04)
CTL E	(C5)	= ENQ	(2D)	(05)
CTL F	(C6)	= ACK	(2E)	(06)
CTL G	(C7)	= BEL	(2F)	(07)
CTL H	(C8)	= BS	(16)	(08)
CTL I	(C9)	= HT	(05)	(09)
CTL J	(D1)	= LF	(25)	(0A)
CTL K	(D2)	= VT	(0B)	(0B)
CTL L	(D3)	= FF	(0C)	(0C)
CTL M	(D4)	= CR	(0D)	(0D)
CTL N	(D5)	= SO	(0E)	(0E)
CTL O	(D6)	= SI	(0F)	(0F)
CTL P	(D7)	= DLE	(10)	(10)
CTL Q	(D8)	= DC1	(11)	(11)
CTL R	(D9)	= DC2	(12)	(12)
CTL S	(E2)	= DC3	(13)	(13)
CTL T	(E3)	= DC4	(3C)	(14)
CTL U	(E4)	= NAK	(3D)	(15)
CTL V	(E5)	= SYN	(32)	(16)
CTL W	(E6)	= ETB	(26)	(17)
CTL X	(E7)	= CAN	(18)	(18)
CTL Y	(E8)	= EM	(19)	(19)
CTL Z	(E9)	= SUB	(3F)	(1A)

Keyboard Control Character Mappings
FIGURE 5

	EBCDIC		EBCDIC	ASCII	
CTL 1	(F1) =	BREAK	(38)	(81)	- CIRCLE C
CTL 2	(F2) =	NOP	(17)	(82)	- IDLE
CTL 3	(F3) =	NO ECHO	(14)	(83)	- RESTORE
CTL 4	(F4) =	ECHO	(23)	(84)	
CTL 5	(F5) =	HIDE YOU INPUT	(24)	(85)	- BYPASS

DATA MARK (80) CANNOT BE ENTERED FROM THE KEYBOARD

THE FOLLOWING 2741 KEYBOARD CHARACTERS DO NOT HAVE A MEANING AS A CONTROL:

```
$ # * % ¢
+ - = _
. , : ;
! | ? ¬
SPACE
BACKSPACE
TAB
```

Keyboard Control Character Mappings
FIGURE 5 (CONTINUTED)

```
•y  0 1 2 3 4 5 6 7 8 9 A B C D E F
x

4   . . . . . . . . . . . . .¢...<.(.+.|.
5   .&. . . . . . . . . . . .!.$.*.).;.¬.
6   .-./. . . . . . . . . . .,.%._.>.?.
7   . . . . . . . . . . . . .:.#.@.'.=."
8   . .a.b.c.d.e.f.g.h.i. . . . . . . .
9   . .j.k.l.m.n.o.p.q.r. . . . . . . .
A   . . .s.t.u.v.w.x.y.z. . . . . . . .
B   . . . . . . . . . . . . . . . . . .
C   . .A.B.C.D.E.F.G.H.I. . . . . . . .
D   . .J.K.L.M.N.O.P.Q.R. . . . . . . .
E   . . .S.T.M.V.W.X.Y.Z. . . . . . . .
F   .0.1.2.3.4.5.6.7.8.9. . . . . . . .
```

Hex Code X'xy' for Characters on a 2741 Terminal

```
••y  0 1 2 3 4 5 6 7 8 9
xx

06                 . . . . . . .
07  . . . . .¢...<.(.+.|.
08  .&. . . . . . . . . .
09  .!.$.*.).;.¬. ./. . .
10  . . . . . .,.%._.
11  .>.?. . . . . . . . .
12  . . .:.#.@.'.=.". .a.
13  .b.c.d.e.f.g.h.i. . .
14  . . . . . .j.k.l.m.n.
15  .o.p.q.r. . . . . . .
16  . . .s.t.u.v.w.x.y.z.
17  . . . . . . . . . . .
18  . . . . . . . . . . .
19  . . . .A.B.C.D.E.F.G.
20  .H.I. . . . . . . .J.
21  .K.L.M.N.O.P.Q.R. . .
22  . . . . . .S.T.U.V.
23  .W.X.Y.Z. . . . . . .
24  .0.1.2.3.4.5.6.7.8.9.
25  . . . . . . .
```

Decimal Code D'xxy' for Characters on a 2741 terminal

HT	X'05' =	D'005'	Horizontal Tab
LC	X'06' =	D'006'	Lower Case
RES	X'14' =	D'020'	Print Restore
NL	X'15' =	D'021'	New Line
BS	X'16' =	D'022'	Back Space
IL	X'17' =	D'023'	Idle
BYP	X'24' =	D'036'	Print Bypass
LF	X'25' =	D'037'	Line Feed
UC	X'36' =	D'054'	Upper Case

Hex Code X'xy' and Decimal Code D'xxy' for 2741 Control Codes

FIGURE 6

```
•y  0 1 2 3 4 5 6 7 8 9 A B C D E F
x

0   . .A.B.C.D.E.F.G.H.I. ...<.(.+.|.
1   .&.J.K.L.M.N.O.P.Q.R. .$.*.).;.¬.
2   .-./.S.T.U.V.W.X.Y.Z. .,.%._.>.?.
3   .0.1.2.3.4.5.6.7.8.9.:.#.@.'.=.".
4   . .A.B.C.D.E.F.G.H.I. ...<.(.+.|.
5   .&.J.K.L.M.N.O.P.Q.R. .$.*.).;.¬.
6   .-./.S.T.U.V.W.X.Y.Z. .,.%._.>.?.
7   .0.1.2.3.4.5.6.7.8.9.:.#.@.'.=.".
8   . .A.B.C.D.E.F.G.H.I. ...<.(.+.|.
9   .&.J.K.L.M.N.O.P.Q.R. .$.*.).;.¬.
A   .-./.S.T.U.V.W.X.Y.Z. .,.%._.>.?.
B   .0.1.2.3.4.5.6.7.8.9.:.#.@.'.=.".
C   . .A.B.C.D.E.F.G.H.I. ...<.(.+.|.
D   .&.J.K.L.M.N.O.P.Q.R. .$.*.).;.¬.
E   .-./.S.T.U.V.W.X.Y.Z. .,.%._.>.?.
F   .0.1.2.3.4.5.6.7.8.9.:.#.@.'.=.".
```

Hex Code X'xy' for Characters on the PN train

```
••y  0 1 2 3 4 5 6 7 8 9
xx

00  . .A.B.C.D.E.F.G.H.I.
01  . ...<.(.+.|.&.J.K.L.
02  .M.N.O.P.Q.R. .$.*.).
03  .;.¬.-./.S.T.U.V.W.X.
04  .Y.Z. .,.%._.>.?.0.1.
05  .2.3.4.5.6.7.8.9.:.#.
06  .@.'.=.". .A.B.C.D.E.
07  .F.G.H.I. ...<.(.+.|.
08  .&.J.K.L.M.N.O.P.Q.R.
09  . .$.*.).;.¬.-./.S.T.
10  .U.V.W.X.Y.Z. .,.%._.
11  .>.?.0.1.2.3.4.5.6.7.
12  .8.9.:.#.@.'.=.". .A.
13  .B.C.D.E.F.G.H.I. ...
14  .<.(.+.|.&.J.K.L.M.N.
15  .O.P.Q.R. .$.*.).;.¬.
16  .-./.S.T.U.V.W.X.Y.Z.
17  . .,.%._.>.?.0.1.2.3.
18  .4.5.6.7.8.9.:.#.@.'.
19  .=.". .A.B.C.D.E.F.G.
20  .H.I. ...<.(.+.|.&.J.
21  .K.L.M.N.O.P.Q.R. .$.
22  .*.).;.¬.-./.S.T.U.V.
23  .W.X.Y.Z. .,.%._.>.?.
24  .0.1.2.3.4.5.6.7.8.9.
25  .:.#.@.'.=.".
```

Decimal Code D'xxy' for Characters on the PN train

FIGURE 7

```
•y  0 1 2 3 4 5 6 7 8 9 A B C D E F
x

0   . . . . . . . . . . . . . . . .
1   . . . . . . . . . . . . . . . .
2   . . . . . . . . . . . . . . . .
3   . . . . . . . . . . . . . . . .
4   . . . . . . . . . . . ¢ . . < . ( . + . | .
5   . & . . . . . . . . . ! . $ . * . ) . ; . ¬ .
6   . - . / . . . . . . . . , . % . _ . > . ? .
7   . . . . . . . . . . . : . # . @ . ' . = . " .
8   . . a . b . c . d . e . f . g . h . i . . { . ≤ . ( . + . ± .
9   . . j . k . l . m . n . o . p . q . r . . } . □ . ) . ± . ¤ .
A   . ~ . ^ . s . t . u . v . w . x . y . z . . └ . ┌ . [ . ≥ . ◦ .
B   . ^ . ¹ . ² . ³ . ⁴ . ⁵ . ⁶ . ⁷ . ⁸ . ⁹ . . ┘ . ┐ . ] . ≠ . — .
C   . . A . B . C . D . E . F . G . H . I . . . . . . .
D   . . J . K . L . M . N . O . P . Q . R . . . . . . .
E   . . . S . T . U . V . W . X . Y . Z . . . . . . .
F   . 0 . 1 . 2 . 3 . 4 . 5 . 6 . 7 . 8 . 9 . . . . . .
```

Hex Code X'xy' for Characters on the TN train

```
••y  0 1 2 3 4 5 6 7 8 9
xx

00   . . . . . . . . . .
01   . . . . . . . . . .
02   . . . . . . . . . .
03   . . . . . . . . . .
04   . . . . . . . . . .
05   . . . . . . . . . .
06   . . . . . . . . . .
07   . . . . ¢ . . . < . ( . + . | .
08   . & . . . . . . . . .
09   . ! . $ . * . ) . ; . ¬ . - . / . . .
10   . . . . . . . . , . % . _ .
11   . > . ? . . . . . . . .
12   . . . : . # . @ . ' . = . " . . a .
13   . b . c . d . e . f . g . h . i . . { .
14   . ≤ . ( . + . ± . . j . k . l . m . n .
15   . o . p . q . r . . } . □ . ) . ± . ¤ .
16   . ~ . ^ . s . t . u . v . w . x . y . z .
17   . . └ . ┌ . [ . ≥ . ◦ . ^ . ¹ . ² . ³ .
18   . ⁴ . ⁵ . ⁶ . ⁷ . ⁸ . ⁹ . . ┘ . ┐ . ] .
19   . ≠ . — . . A . B . C . D . E . F . G .
20   . H . I . . . . . . . . J .
21   . K . L . M . N . O . P . Q . R . . .
22   . . . . . . . S . T . U . V .
23   . W . X . Y . Z . . . . . . .
24   . 0 . 1 . 2 . 3 . 4 . 5 . 6 . 7 . 8 . 9 .
25   . . . . . . . .
```

Decimal Code D'xxy' for Characters on the TN train

FIGURE 8